

~~UNCLASSIFIED~~

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

FILE FILE COPY

2

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

3. DISTRIBUTION/AVAILABILITY OF REPORT

Approved for public release;
distribution unlimited.

AD-A216 511

5. MONITORING ORGANIZATION REPORT NUMBER(S)

AFOSR-TR-89-1863

6a. NAME OF FUNDING/SPONSORING ORGANIZATION
West Virginia University

6b. OFFICE SYMBOL
(If applicable)

7a. NAME OF MONITORING ORGANIZATION

AFOSR

6c. ADDRESS (City, State, and ZIP Code)

Cun-Quan Zhang, Mathematics Department
West Virginia University
Morgantown, WV 26506

7b. ADDRESS (City, State, and ZIP Code)

Bldg. 410
Bolling AFB, DC 20332-6448

8a. NAME OF FUNDING/SPONSORING ORGANIZATION

Air Force
Office of Scientific Research

8b. OFFICE SYMBOL
(If applicable)

NN

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

AFOSR-89-0068

8c. ADDRESS (City, State, and ZIP Code)

AFOSR/NM
Building 410
Bolling AFB, DC 20332-6448

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
AFOSR89-0068	2304	A8	

11. TITLE (Include Security Classification)

A New Parallel Add

12. PERSONAL AUTHOR(S)

Cun-Quan Zhang

13a. TYPE OF REPORT

Final Technical

13b. TIME COVERED

FROM 11/88 TO 10/89

14. DATE OF REPORT (Year, Month, Day)

89/10/10

15. PAGE COUNT

12

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

FIELD	GROUP	SUB-GROUP

A New Parallel Add

To the m place.

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

A new parallel add is introduced in this paper which consists of $(2^m - 1)(m+3) + 1$ 3-input modules and costs $m+1$ time units when processing a sum of two binary numbers of length at most 2^m .

DTIC
ELECTE
JAN 05 1990
S D C S D

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT

UNCLASSIFIED/UNLIMITED SAME AS RPT. DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION

22a. NAME OF RESPONSIBLE INDIVIDUAL

Dr. N. Glassman

22b. TELEPHONE (Include Area Code)

(202) 767-5026

22c. OFFICE SYMBOL

NN

DD Form 1473, JUN 86

Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

90 01 04 162

A NEW PARALLEL ADD

Cun-Quan Zhang*
Department of Mathematics
West Virginia University
Morgantown
West Virginia 26506 USA

ABSTRACT

A new parallel add is introduced in this paper which consists of $(2^m-1)(m+3)+1$ 3-input modules and costs $m+1$ time units when processing a sum of two binary numbers of length at most 2^m .



Accesion For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution / _____	
Availability Codes	
Dist	Avail and/or Special
A-1	

*. This research was partially supported by AFOSR under the grant 89-0068

The adds are basic parts of computers. Many different models of adds have been introduced. Some needs fewer modules and is more time consuming in processing, while some needs more modules and runs faster. Two typical models of adds are series and parallel adds (see [1], pp259). When processing the sum of two binary numbers with 2^m bits, the circuits of the series add consists of 2^m 3-input-modules and costs 2^m time units in processing. The parallel adds can save time in processing which costs only $m+1$ time units. But it consists of $\frac{5}{2}(3^m)$ 3-input modules. In this paper, a new model of parallel add is introduced which needs $(2^m-1)(m+3)+1$ 3-input modules and cost $m+1$ time units in processing a sum of two binary numbers with 2^m bits. It is obvious that $(2^m-1)(m+3)+1 \leq \frac{5}{2}(3^m)$.

Let the binary inputs be $x=x_nx_{n-1}\dots x_1$ and $y=y_ny_{n-1}\dots y_1$ and let $n=2^m$. For the sake of convinience, a sequence of i -th, $(i+1)$ -th, ..., j -th bits of x is denoted by $x_{\{j,\dots,i\}}$.

S 1. PRIMARY MODULES OF ADD.

The new add consists of two primary modules, *primary add module* and *primary selector*.

The primary add has three inputs x , y , z and two outputs s , c which is represented in the diagram by

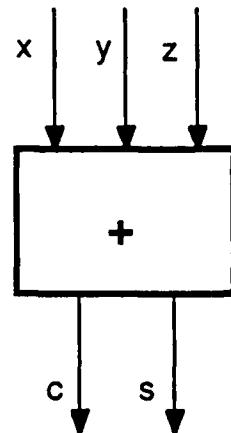


fig. 1

The function of the primary add is $x+y+z=cs$ (cs is a binary number with the first bit s and the second bit c) where x , y and z are input adders and s is the first bit of the sum and c is the carrier (the second bit of the sum). The table of the output cs of the primary add is,

when $z=0$		when $z=1$	
$x=0$	$x=1$	$x=0$	$x=1$
$y=0$	00	01	01
$y=1$	01	10	10

when $z=0$		when $z=1$	
$x=0$	$x=1$	$x=0$	$x=1$
$y=0$	01	10	11
$y=1$	10	11	11

Table 1

The primary selector has two input i_0, i_1 , one select input c_μ and one output o_μ which is represented in the diagram by

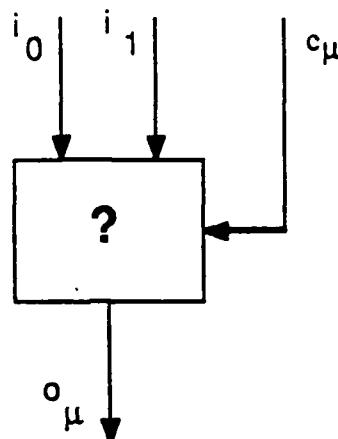


fig. 2

The function of the primary selector is

$$o_\mu = \begin{cases} i_0 & \text{if } c_\mu = 0 \\ i_1 & \text{if } c_\mu = 1 \end{cases}$$

S 2. THE FIRST LEVEL OF AN ADD

The first level of an add is denoted by $\text{Add}(1)$ and consists of two primary adds and two primary selectors. The inputs of $\text{Add}(1)$ are x_i and y_i which are the i -th bits of the input adders. It has two pairs of outputs $\{c_{\mu,i}, s_{\mu,i}\}$ for $\mu=0$ or 1, each of which is with the assumption that the previous carrier (at $(i-1)$ -th position) is μ (for $\mu=0$ or 1), where $c_{\mu,i}, s_{\mu,i} = x_i + y_i + \mu$, $c_{\mu,i}$ is the carrier and $s_{\mu,i}$ is the sum at i -th position. The structure of $\text{Add}(1)$ is illustrated as the follows.

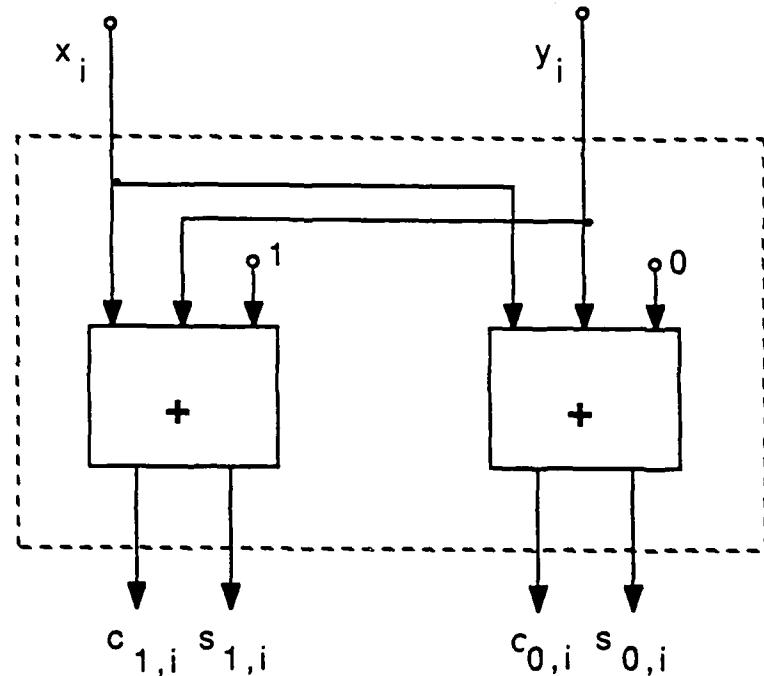


fig. 3

An Add(1) is represented in diagrams by

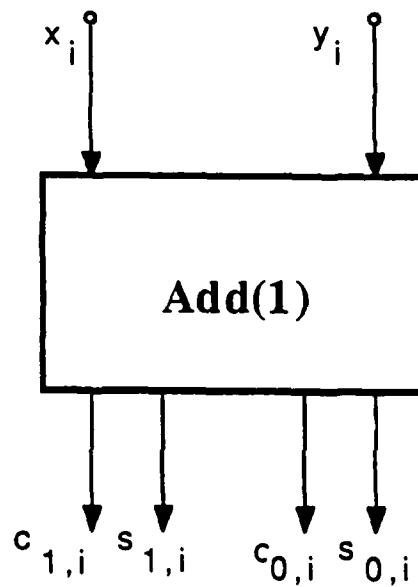


fig. 4

S 3. THE SECOND LEVEL AND THE M-TH LEVEL OF THE ADD

Add(m) is a device which has the ability of processing the sum of two binary numbers of length at most 2^{m-1} with assumptions that the previous carriers is 1 or 0.

The m -th level of add is denoted by Add(m) and is represented in diagram by

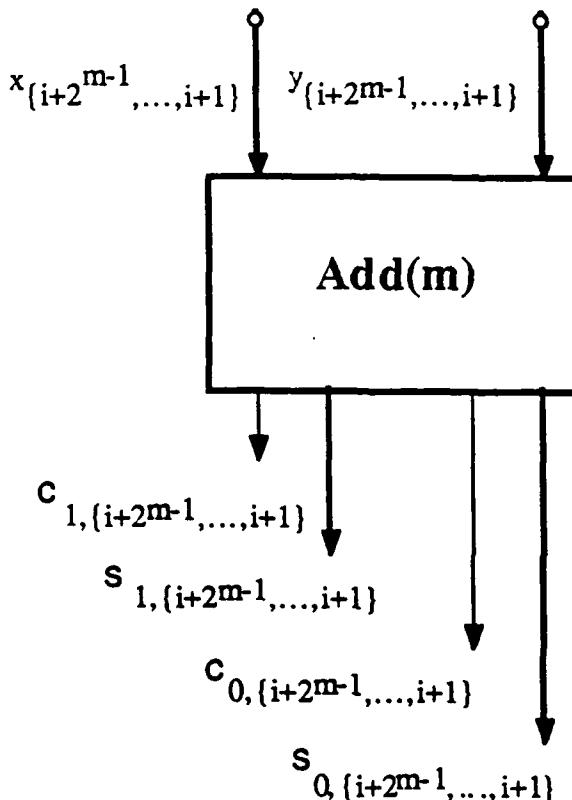


fig. 5

where $x_{\{i+2^{m-1}, \dots, i+1\}}$ and $y_{\{i+2^{m-1}, \dots, i+1\}}$ are the sequences of the $(i+1)$ -th, ..., $(i+2^{m-1})$ -th bits of the inputs x and y , and $c_{\mu,\{i+2^{m-1}, \dots, i+1\}}$, $s_{\mu,\{i+2^{m-1}, \dots, i+1\}}$ are the carrier and the sum of $x_{\{i+2^{m-1}, \dots, i+1\}}$ and $y_{\{i+2^{m-1}, \dots, i+1\}}$ with the assumption that the i -th carrier is μ ($\mu=1$ or 0). That is,

$$c_{\mu,\{i+2^{m-1}, \dots, i+1\}} s_{\mu,\{i+2^{m-1}, \dots, i+1\}} = x_{\{i+2^{m-1}, \dots, i+1\}} + y_{\{i+2^{m-1}, \dots, i+1\}} + \mu,$$

where $c_{\mu,\{i+2^{m-1}, \dots, i+1\}}$ is a single bit and $s_{\mu,\{i+2^{m-1}, \dots, i+1\}}$ is a sequence of 2^{m-1} bits.

The structure of the second level of an add is illustrated as the follows.

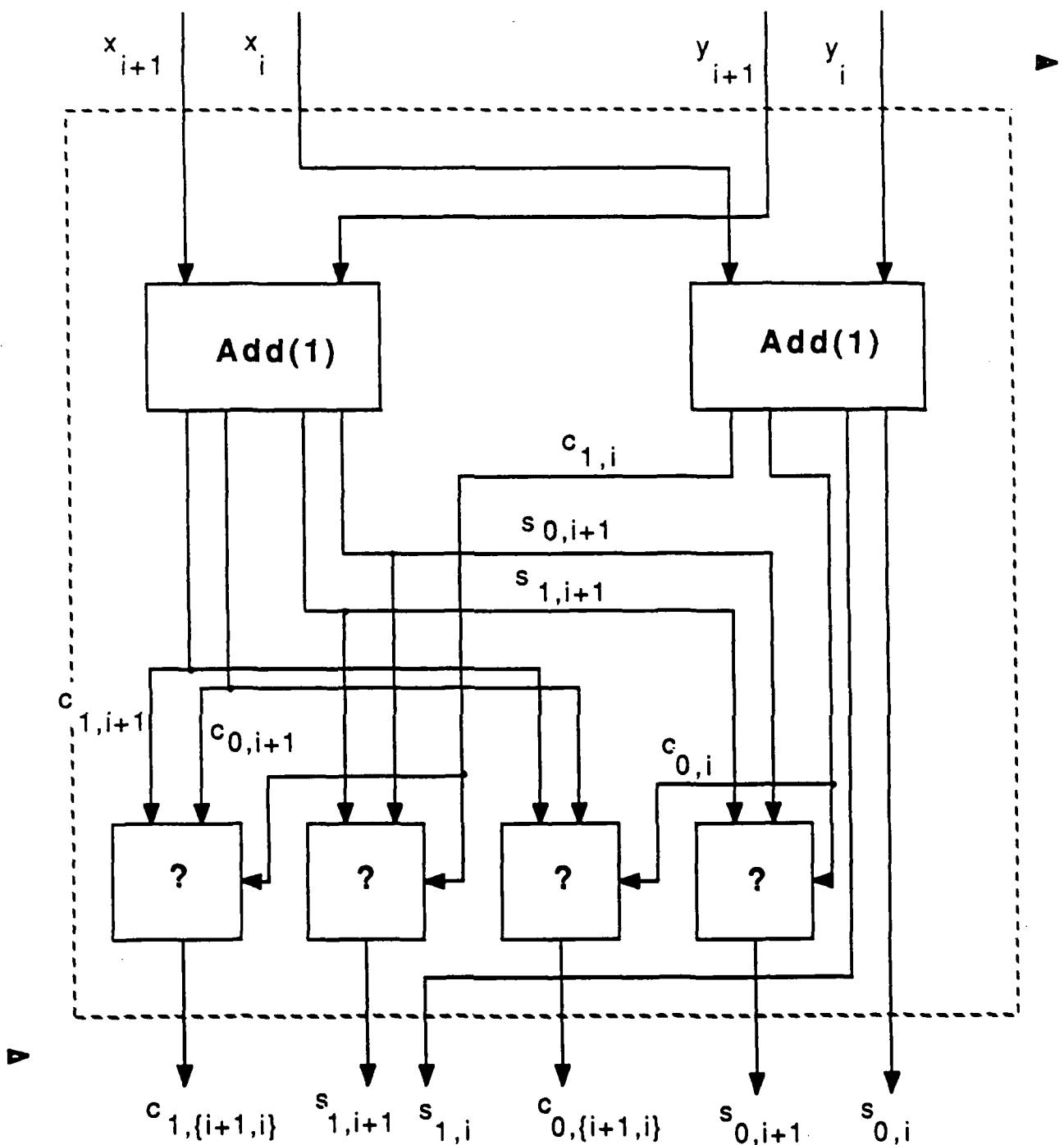


fig. 6

Where the output $s_{\mu,i+1}s_{\mu,i}$ (which will be denoted by $s_{\mu,(i+1,i)}$), is the sum of $x_{(i+1,i)}$ and $y_{(i+1,i)}$ at i -th and $(i+1)$ -th positions, and $c_{\mu,(i+1,i)}$ is the carrier at the $(i+1)$ -th position with the assumption that the previous carrier from the $(i-1)$ -th position is μ ($\mu=1$ or 0).

For the sake of convenience, we simply draw a bunch of parallel wires in diagram

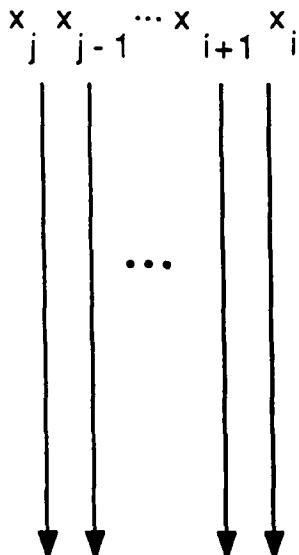


fig. 7

by a bold line

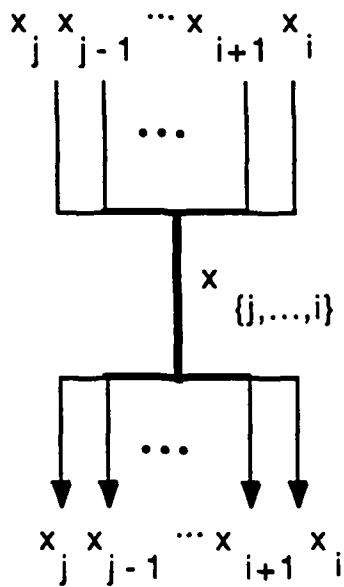


fig. 8

Thus the diagram of an Add(2) can be drawn as the follows,

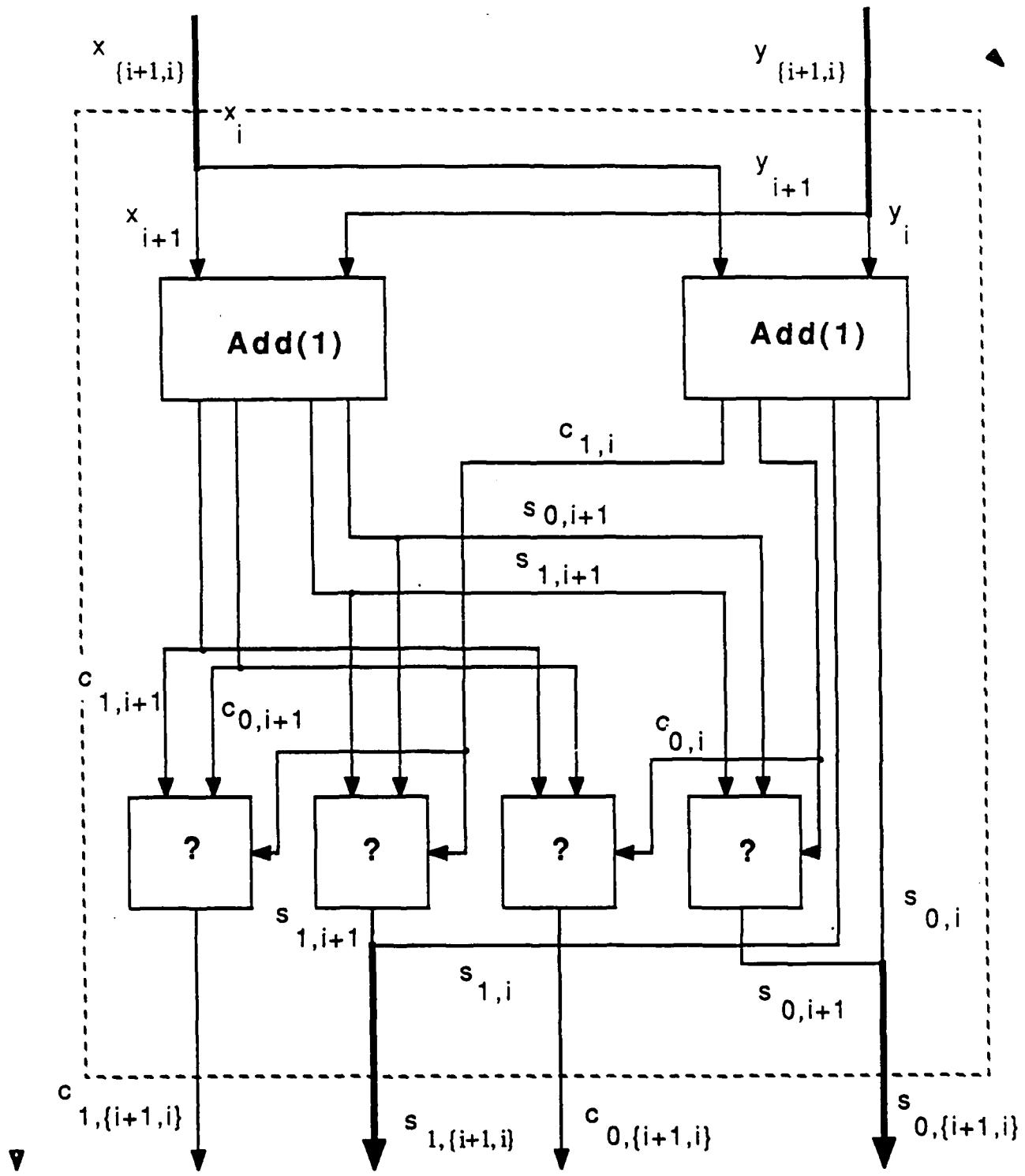


fig. 6'

The $(m+1)$ -th level of an add, $\text{Add}(m+1)$, is constructed recursively as follows.

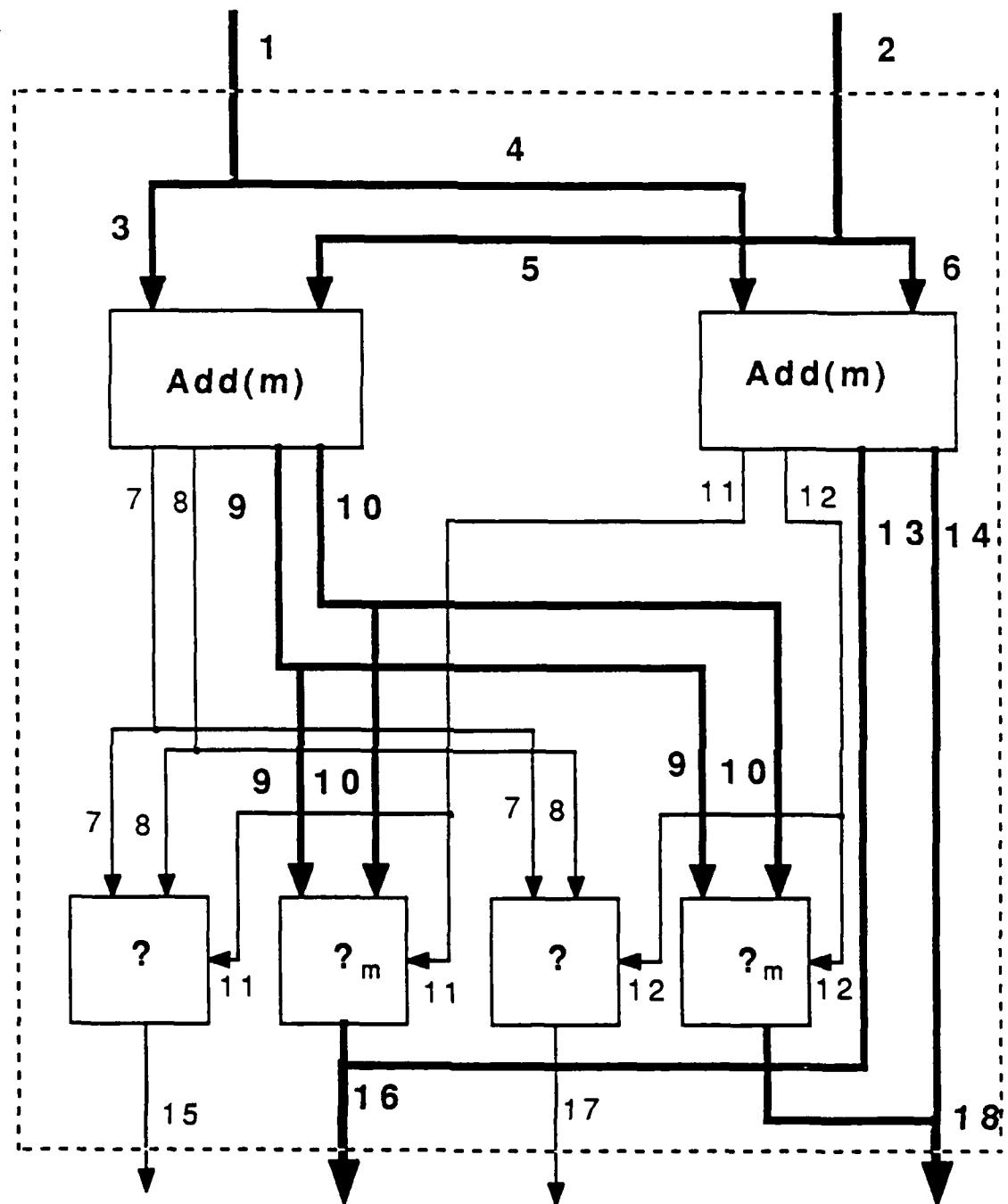


fig. 9

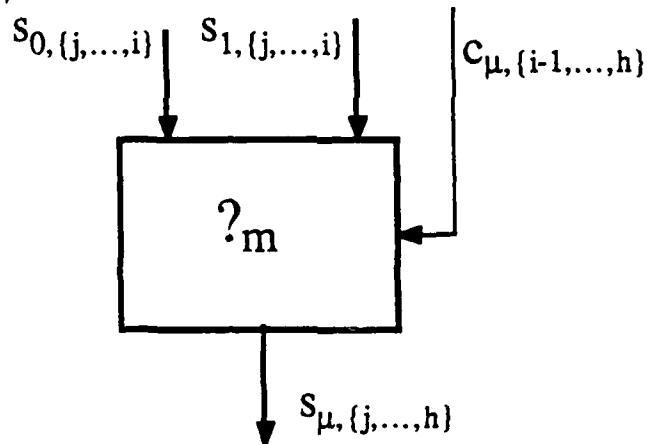
The wires in fig. 9 are described in the following table.

<u>No. of the wire</u>	<u>Description</u>
1.	$X\{i+2m, \dots, i+1\},$
2.	$Y\{i+2m, \dots, i+1\},$

3. $X\{i+2m, \dots, i+2m-1, 1\},$
 4. $Y\{i+2m, \dots, i+2m-1, 1\},$
 5. $X\{i+2m-1, \dots, i+1\},$
 6. $Y\{i+2m-1, \dots, i+1\},$
 7. $C_0, \{i+2m, \dots, i+2m-1, 1\},$
 8. $C_1, \{i+2m, \dots, i+2m-1, 1\},$
 9. $S_0, \{i+2m, \dots, i+2m-1, 1\},$
 10. $S_1, \{i+2m, \dots, i+2m-1, 1\},$
 11. $C_0, \{i+2m-1, \dots, i+1\},$
 12. $C_1, \{i+2m-1, \dots, i+1\},$
 13. $S_0, \{i+2m-1, \dots, i+1\},$
 14. $S_1, \{i+2m-1, \dots, i+1\},$
 15. $C_0, \{i+2m, \dots, i+1\},$
 16. $S_0, \{i+2m, \dots, i+1\},$
 17. $C_1, \{i+2m, \dots, i+1\},$
 18. $S_1, \{i+2m, \dots, i+1\},$

Table 2

In the fig. 9,



is a series of primary selectors illustrated as following

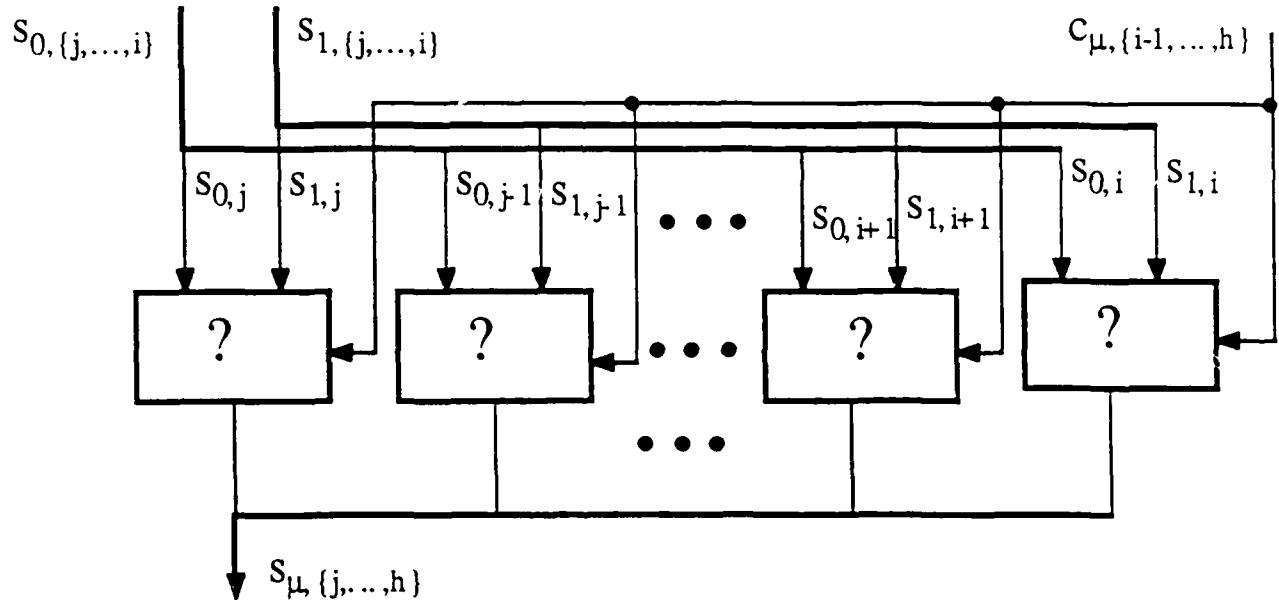


fig. 10

§ 4. TIME COST AND NUMBER OF MODULES

THEOREM 1.

The time cost of processing a sum of two binary numbers with 2^{m-1} bits in an Add(m) is m and the number of 3-input modules in an Add(m) is $2^{m+1} + (m-1)2^{m-1} - 2$.

PROOF.

Prove by induction on m .

Let the number 3-input modules of an Add(m) be $N(m)$. It is trivial if $m=1$. And assume that

$$N(m-1) = 2^m + (m-2)2^{m-2} - 2.$$

By the recursive construction of the add, we have that
 $N(m) = 2\{\text{number of modules in Add}(m-1) + \text{number of primary selectors}$
 in m -th level selector}

$$\begin{aligned} &= 2N(m-1) + 2(1+2^{m-2}) \\ &= 2[2^m + (m-2)2^{m-2} - 2] + 2(1 + 2^{m-2}) \\ &= 2^{m+1} + (m-1)2^{m-1} - 2 \end{aligned}$$

Assume that the cost of processing a sum of binary numbers with 2^{m-2} bits in an Add($m-1$) is $m-1$. From the structure of an Add(m), it costs only one more time unit to pass through the selectors succeeding the Add($m-1$)'s. Thus the time cost is m for processing a sum in an Add(m)

###

Let $\lceil r \rceil$ be the least integer not less than r . It is obvious that the processing of a sum of two binary numbers of

lengths at most n only can be done by an $\text{Add}(m)$ with
 $m \geq \lceil \log_2 n \rceil + 1$. Note that the integer n might not be a power
of 2. Let $n \leq 2^{m-1}$. Thus replacing $m-1$ and 2^{m-1} by $\lceil \log_2 n \rceil$
and $2n$, respectively, in the Theorem 1, we will have the
following corollaries.

COROLLARY 2.

The time cost of processing a sum of two binary numbers
of lengths n is at most $\lceil \log_2 n \rceil + 1$.

COROLLARY 3.

The number of 3-input modules of an add for processing
binary inputs of length n is at most $2n(4 + \lceil \log_2 n \rceil)$.

REMARK.

Since there is no carrier before the first bit or the
carrier before the first bit is always considered as zero,
each $\text{Add}(m)$ dealing with the first several bits don't need to
have the wires 12, 14, 17 and 18 (see fig. 9 and table 2).
Thus $1+2^{m-2}$ primary selectors can be saved in the m -th level,
and one primary add can be saved in the first level of such
add. By considering that, the numbers of 3-input modules can
be reduced to $2^{m+1} + (m-2)2^{m-1} - m-1$. That is, the number of
3-input-modules in an add for processing the sum of two
binary inputs of length n is at most

$$(2n-1)(3 + \lceil \log_2 n \rceil) + 5.$$

REFERENCES

1. H.B. Laufer, Discrete Mathematics and Applied Modern Algebra, Prindle, Weber & Schmidt, Boston, (1984).